

Experiment No. 8

Aim: Write a test plan and test case for any software application

Part A:

Introduction

Test plan: A document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process.

TEST PLAN TEMPLATE

The format and content of a software test plan vary depending on the processes, standards, and test management tools being implemented. Nevertheless, the following format, which is based on IEEE standard for software test documentation, provides a summary of what a test plan can/should contain.

Test Plan Identifier:

- Provide a unique identifier for the document. (Adhere to the Configuration Management System if you have one.)

Introduction:

- Provide an overview of the test plan.
- Specify the goals/objectives.
- Specify any constraints.

References:

- List the related documents, with links to them if available, including the following:
 - Project Plan
 - Configuration Management Plan

Test Items:

- List the test items (software/products) and their versions.

Features to be tested:

- List the features of the software/product to be tested.
- Provide references to the Requirements and/or Design specifications of the features to be tested

Features Not to Be Tested:

- List the features of the software/product which will not be tested.
- Specify the reasons these features won't be tested.

Approach:

- Mention the overall approach to testing.
- Specify the testing levels [if it's a Master Test Plan], the testing types, and the testing methods [Manual/Automated; White Box/Black Box/Gray Box]

Item Pass/Fail Criteria:

- Specify the criteria that will be used to determine whether each test item (software/product) has passed or failed testing.

Suspension Criteria and Resumption Requirements:

- Specify criteria to be used to suspend the testing activity.
- Specify testing activities which must be redone when testing is resumed.

Test Deliverables:

- List test deliverables, and links to them if available, including the following:
 - Test Plan (this document itself)
 - Test Cases
 - Test Scripts
 - Defect/Enhancement Logs
 - Test Reports

Test Environment:

- Specify the properties of test environment: hardware, software, network etc.
- List any testing or related tools.

Estimate:

- Provide a summary of test estimates (cost or effort) and/or provide a link to the detailed estimation.

Schedule:

- Provide a summary of the schedule, specifying key test milestones, and/or provide a link to the detailed schedule.

Staffing and Training Needs:

- Specify staffing needs by role and required skills.
- Identify training that is necessary to provide those skills, if not already acquired.

Responsibilities:

- List the responsibilities of each team/role/individual.

Risks:

- List the risks that have been identified.
- Specify the mitigation plan and the contingency plan for each risk.

Assumptions and Dependencies:

- List the assumptions that have been made during the preparation of this plan.
- List the dependencies.

Approvals:

- Specify the names and roles of all persons who must approve the plan.
- Provide space for signatures and dates. (If the document is to be printed.)

TEST PLAN GUIDELINES

- Make the plan concise. Avoid redundancy and superfluosness. If you think you do not need a section that has been mentioned in the template above, go ahead and delete that section in your test plan.
- Be specific. For example, when you specify an operating system as a property of a test environment, mention the OS Edition/Version as well, not just the OS Name.
- Make use of lists and tables wherever possible. Avoid lengthy paragraphs.
- Have the test plan reviewed a number of times prior to baselining it or sending it for approval. The quality of your test plan speaks volumes about the quality of the testing you or your team is going to perform.
- Update the plan as and when necessary. An out-dated and unused document stinks and is worse than not having the document in the first place.

Part B:

Prepare a test plan and test cases for any selected application.

This document should define all the procedures and activities required to prepare for testing of the functionalities of the system. The objectives of the test plan are to define the activities to perform testing, define the test deliverables documents and to identify the various risks and contingencies involved in testing.

Test Plan Identifier: Language Understanding NLP Plan_v1.0

Introduction:

This test the procedures and activities to evaluate the functionalities the Language Understanding software Natural Language Processing (NLP). The main is to ascertain the's adeptness in comprehending and natural language input accurately.

References:

- Project Plan: Outlines overall project goals and timelines.
- Configuration Management Plan: Provides guidelines for managing software configurations.

Test Items:

- Language Understanding NLP Software v1.0: The software version under test.

Features to be tested:

1. Natural Language Input Parsing and Understanding:
 - Evaluate the software's ability to parse and comprehend natural language input accurately.
2. Intent Recognition:
 - Assess the software's capability to recognize the intended action or purpose behind the input.
3. Entity Extraction:
 - Validate the software's proficiency in identifying and extracting relevant entities from the input.
4. Context Management:
 - Verify the software's aptitude in maintaining contextual relevance during conversations or interactions.
5. Response Generation:
 - Ensure the software generates appropriate and contextually relevant responses based on the input received.

Features Not to Be Tested:

- Integration with External Systems: Testing integration points with external systems is out of scope for this testing phase.
- High-level Algorithm Testing: Detailed testing of underlying algorithms is not included in this plan and is typically performed during the development phase.

Approach:

- Overall approach: Black box testing method will be employed to evaluate the software's functionalities.
- Testing levels: System testing level will be focused on for comprehensive evaluation.
- Testing types: Functional testing will be the primary testing type utilized.

- Testing methods: Manual testing will be conducted to assess the software's performance.

Item Pass/Fail Criteria:

- Parsing and Understanding:
 - **Pass:** Natural language input is interpreted accurately with minimal errors.
 - **Fail:** Inaccurate interpretation or significant deviation from expected output.
- Intent Recognition:
 - **Pass:** Correct identification of the intended action or purpose behind the input.
 - **Fail:** Incorrect or inconsistent recognition of intent.
- Entity Extraction:
 - **Pass:** Relevant entities are extracted accurately from the input.
 - **Fail:** Missing or incorrect extraction of entities.
- Context Management:
 - **Pass:** Software maintains contextual relevance throughout the interaction.
 - **Fail:** Loss of context or irrelevant responses.
- Response Generation:
 - **Pass:** Appropriate and contextually relevant responses are generated.
 - **Fail:** Inappropriate or irrelevant responses.

Suspension Criteria and Resumption Requirements:

- **Suspension Criteria:** Testing activities may be suspended if critical defects affecting core functionalities are identified.
- **Resumption Requirements:** Testing will resume once defects are resolved and verified by the development team.

Test Deliverables:

- Test Plan (this document)
- Test Cases
- Test Scripts
- Defect/Enhancement Logs
- Test Reports

Test Environment:

- **Hardware:** Standard PCs with sufficient processing power and memory.
- **Software:** Operating systems compatible with the Language Understanding NLP software.
- **Network:** No specific network requirements.
- **Testing Tools:** NLP testing frameworks may be utilized if available.

Estimate:

- **Effort Estimate:** Approximately 100 person-hours will be allocated for testing activities.

Schedule:

- **Test Plan Review:** [Date]
- **Test Case Creation:** [Date]
- **Test Execution:** [Date]
- **Test Report Generation:** [Date]

Staffing and Training Needs:

- **Staffing Needs:** Testers with experience in NLP testing will be assigned.
- **Training Needs:** Testers may undergo training to familiarize themselves with NLP concepts and testing techniques if necessary.

Responsibilities:

- **Testers:** Responsible for executing test cases, reporting defects, and collaborating with developers.
- **Developers:** Responsible for fixing defects, providing clarification on requirements, and assisting testers during testing activities.

Risks:

- **Data Quality:** Insufficient or inaccurate training data may lead to poor performance.
- **Mitigation Plan:** Thorough data preprocessing and validation will be conducted to ensure data quality.
- **Contingency Plan:** Synthetic data or augmented training data may be utilized if necessary- **Algorithm Complexity:** Complex algorithms may be prone to errors.
- **Mitigation Plan:** Comprehensive code reviews and unit testing will be conducted to mitigate algorithmic risks.
- **Contingency Plan:** Additional resources will be allocated for debugging and optimization if required.

Assumptions and Dependencies:

- **Assumption:** Suitable test data for training and testing purposes will be available.
- **Dependency:** Availability of required hardware and software resources for testing purposes.

Approvals:

This test plan requires approval from the following:

- [Project Manager], Project Manager
- [Development Lead], Development Lead
- [Space for signatures and dates]

Test Cases

Test Case ID	Description	Preconditions	Test Steps	Expected Result	Pass/Fail Criteria
TC001	Verify Natural Language Input Parsing and Understanding	Language Understanding NLP software is running	1. Enter valid natural language input. 2. Verify if the software accurately parses and understands the input.	Software interprets the input correctly.	Pass if the input is interpreted accurately, fail otherwise.
TC002	Verify Intent Recognition	Language Understanding NLP software is running	1. Enter natural language input with a specific intent. 2. Verify if the software recognizes the correct intent.	Software identifies the correct intent.	Pass if the intent is recognized correctly, fail otherwise.
TC003	Verify Entity Extraction	Language Understanding NLP software is running	1. Enter natural language input with entities. 2. Verify if the software correctly identifies and extracts relevant entities.	Relevant entities are extracted accurately.	Pass if entities are extracted accurately, fail otherwise.
TC004	Verify Context Management	Language Understanding NLP software is running	1. Engage in a conversation or interaction with contextual cues. 2. Verify if the software maintains contextual relevance in responses.	Software maintains contextual relevance.	Pass if context is maintained throughout, fail otherwise.
TC005	Verify Response Generation	Language Understanding NLP software is running	1. Input natural language queries or statements. 2. Verify if the software	Appropriate responses are generated.	Pass if responses are relevant and contextually appropriate, fail otherwise.

			generates appropriate and contextually relevant responses.		
TC006	Verify Handling of Ambiguous Inputs	Language Understanding NLP software is running	1. Input natural language queries with ambiguous or unclear meanings. 2. Verify if the software handles ambiguous inputs appropriately.	Software provides clarification or asks for clarification.	Pass if ambiguous inputs are handled appropriately, fail otherwise.
TC007	Verify Error Handling	Language Understanding NLP software is running	1. Input erroneous or malformed natural language input. 2. Verify if the software handles errors gracefully without crashing.	Software provides error message or fallback response.	Pass if errors are handled gracefully, fail otherwise.
TC008	Verify Multi-language Support	Language Understanding NLP software is running	1. Input natural language queries in different languages. 2. Verify if the software accurately processes and responds to multi-language inputs.	Software accurately processes and responds to multi-language inputs.	Pass if multi-language support is accurate, fail otherwise.